UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/769,270	01/26/2001	Seiji Shima	520.39555X00	4351

20457 7590 07/27/2004

ANTONELLI, TERRY, STOUT & KRAUS, LLP
1300 NORTH SEVENTEENTH STREET
SUITE 1800
ARLINGTON, VA 22209-9889

EXAMINER	
PHAM, HUNG Q	
ART UNIT	**PAPER NUMBER**
2172	14

DATE MAILED: 07/27/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No. 09/769,270	Applicant(s) SHIMA ET AL.
	Examiner HUNG Q PHAM	Art Unit 2172

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 May 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 05/25/2004 has been entered.

Response to Arguments

2. Applicant's arguments based on the amendment of claims 1, 4 and 9, see pages 8-9, filed on 05/25/2004, with respect to the rejection under USC § 112, first paragraph, have been fully considered and are persuasive. The rejection under USC § 112 of claims 1, 4 and 9 has been withdrawn.

3. Applicant's arguments of new added features in claims 1, 4, 6 and 9, on pages 12-13 with respect to the rejection under USC § 103, will be detailed as in the following action.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claims 1-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over O'Connor [USP 6,564,228 B1] in view of Dang et al. [USP 5,446,855].

Regarding to claim 1, O'Connor teaches a network file system and method wherein a storage area network Universal File System allows any host in a heterogeneous based storage area network to read or write data as if in its native

format (Abstract). As shown in FIG. 4 is a storage area network includes Universal File System (UVFS) storage devices 420, type A storage device 422, type B storage device 424, type A hosts 402 and 406, and type B hosts 404 and 408. Type A host 402 and type B host 404 both include universal file system mechanisms 410 and 412, respectively. The file system of the UVFS storage devices 420 is not compatible with the file systems of type A hosts 402 and 406, or type B hosts 404 and 406. Also, the file system of type A hosts 402 and 406 is not compatible with the file systems of type B hosts 404 and 408 (FIG. 4, Col. 5, lines 14-31). To enable a host to utilize a universal file system, software may be installed as part of the operating system of a host, which allows it to mount the universal file system. Once mounted, data may be read from and written to the file system. When a client mounts a directory on a server, that directory and subdirectories become part of the client's directory hierarchy. Each platform may have its own enabling software package. With a universal file system, each platform need only create a package for accessing the universal file system and can be assured of being able to share data with other platforms, which are enabled in like manner (Col. 6, lines 23-38). As seen, once mounted to the universal file system by utilizing the software installed as part of the operating system of a host, a directory, subdirectory, or a file under a directory or subdirectory in a particular file system as ***a unit of data specific to an operating system***, such as type A, is converted to a type that common to the UVFS file system as ***unit of data common to said storages***. In other words, the O'Connor software performs the function of ***a converter facility, included in said host, for***

converting a unit of data specific to an operating system (OS) on said host into a unit of data common to said storages. Referring back to the technique of mounting the host file system on the server, once mounted, data may be read from and written to the file system. When a client mounts a directory on a server, that directory and subdirectories become part of the client's directory hierarchy (Col. 6, lines 23-38). As shown in ***the data transfer networks*** of FIG. 4, UVFS mechanism 410 is configured such that type A host 402 sees data stored on the UVFS storage devices 420 as if it were stored in a format compatible with its own type A format (Col. 5, lines 47-53). A universal permissions scheme modeled after the Unix scheme is used in a universal file system to ensure data security and integrity. For example, when configuring a host for a SAN universal file system, a listing of the permissions mask for a file may be "Urwxr-x-x". In this case, the first character indicates this is a universal file system and should be treated as such. Each user may have one or more of the following permissions: read access, write access, or execute access. When a user attempts to access a file, the operating system first identifies which type of user is making the request, then checks the permissions for that user to determine if access is granted (Col. 6, line 48-Col. 7, line 18). Thus, a type A host 402 of FIG. 4 as ***a host from one of said storages*** accesses a file in Unix scheme modeled UVFS 420 by conventionally specifying the file name such as "Urwxr-x-x" as ***a name of a unit of data common to said storages from said host***, upon the file name reception, the process of retrieving of a file, or ***readout***, is conducted by checking the permissions for that user to determine if access is granted, then displaying the file in a format compatible with

its own type for reading, modifying or executing. In short, this technique indicates ***a management facility, connected to said data transfer networks, for receiving a name of a unit of data common to said storages from said host and managing a readout of said unit of data common to said storages in responds to said unit name received from said host from one of said storages***. FIG. 12 is ***a*** RAID ***storage device for storing data***, which includes disk arrays 1100A and 1100B. O'Connor fails to teach ***a controller for controlling data sent from said host through said data transfer network so as to assign said data to a virtual space and store said data assigned to the virtual space in said storage device***. Dang teaches a system for managing I/O request directed to a disk array (Dang, abstract). As shown in Dang FIG. 2 is a RAID storage device 26 with a plurality of disk array. As shown in FIG. 1, when the processing unit 12 executes an instruction within the application program 19 corresponding to an I/O operation, control is transferred to the operating system 18. The operating system 18 creates a virtual disk I/O request corresponding to the I/O operation requested by the application program, and stores the virtual disk I/O request in RAM 14 as ***a virtual space***. A virtual disk I/O request can either be a virtual disk write request, for which data is to be written to the disk array 26, or a virtual disk read request requiring a disk array data read operation (Dang, Col. 6, lines 39-57). The request is broken into one or more sub-requests (Dang, Col. 7, lines 15-16), and stored in a pending queue (Dang, Col. 8, lines 14-15). If the request is a write request (Dang, FIG. 7A, step 204), the data are written into appropriate disk drives of disk array 26 as ***storage device*** (Dang, Col. 7, lines 44-53). As seen, the Dang technique indicates ***a controller for controlling data sent from said host***

through said data transfer network so as to assign said data to a virtual space and store said data assigned to the virtual space in said storage device. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the O'Connor network file system by including a controller for assigning data to a virtual space before storing in storage device as taught by Dang in order to minimize the amount of time required for performing read write operation in RAID storage device of storage area network Universal File System.

Regarding to claim 4, O'Connor teaches a network file system and method wherein a storage area network Universal File System allows any host in a heterogeneous based storage area network to read or write data as if in its native format (Abstract). As shown in FIG. 4 is a storage area network includes Universal File System (UVFS) storage devices 420, type A storage device 422, type B storage device 424, type A hosts 402 and 406, and type B hosts 404 and 408. Type A host 402 and type B host 404 both include universal file system mechanisms 410 and 412, respectively. The file system of the UVFS storage devices 420 is not compatible with the file systems of type A hosts 402 and 406, or type B hosts 404 and 406. Also, the file system of type A hosts 402 and 406 is not compatible with the file systems of type B hosts 404 and 408 (FIG. 4, Col. 5, lines 14-31). To enable a host to utilize a universal file system, software may be installed as part of the operating system of a host, which allows it to mount the universal file system. Once mounted, data may be read from and written to the file system. When a client mounts a directory on a server, that directory and

subdirectories become part of the client's directory hierarchy. Each platform may have its own enabling software package. With a universal file system, each platform need only create a package for accessing the universal file system and can be assured of being able to share data with other platforms, which are enabled in like manner (Col. 6, lines 23-38). As seen, once mounted to the universal file system by utilizing the software installed as part of the operating system of a host, a directory, subdirectory, or a file under a directory or subdirectory in a particular file system, such as type A as ***a first format having a file format specific to an operating system on said host***, is converted to a type that common to the UVFS file system as ***a second format having file format common to said storages***. In other words, the O'Connor software performs the function of ***a converter facility, included in said host, for converting files in a first format having a file format specific to an operating system on said host into files in a second format having file format common to said storages***. Referring back to the technique of mounting the host file system on the server, once mounted, data may be read from and written to the file system. When a client mounts a directory on a server, that directory and subdirectories become part of the client's directory hierarchy (Col. 6, lines 23-38). As shown in ***the data transfer networks*** of FIG. 4, UVFS mechanism 410 is configured such that type A host 402 sees data stored on the UVFS storage devices 420 as if it were stored in a format compatible with its own type A format (Col. 5, lines 47-53). A universal permissions scheme modeled after the Unix scheme is used in a universal file system to ensure data security and integrity. For example, when configuring a host for a SAN universal file

system, a listing of the permissions mask for a file may be "Urwxr-x-x". In this case, the first character indicates this is a universal file system and should be treated as such. Each user may have one or more of the following permissions: read access, write access, or execute access. When a user attempts to access a file, the operating system first identifies which type of user is making the request, then checks the permissions for that user to determine if access is granted (Col. 6, line 48-Col. 7, line 18). Thus, a type A host 402 of FIG. 4 as ***a host from one of said storages*** accesses a file in Unix scheme modeled UVFS 420 by conventionally specifying the file name such as "Urwxr-x-x" as ***a name of a unit of data common to said storages from said host***, upon the file name reception, the process of retrieving of a file, or ***readout***, is conducted by checking the permissions for that user to determine if access is granted, then displaying the file in a format compatible with its own type for reading, modifying or executing. In short, this technique indicates ***a management facility, connected to said data transfer networks, for receiving a name of a unit of data common to said storages from said host and managing a readout of said unit of data common to said storages in responds to said unit name received from said host from one of said storages***. FIG. 12 is ***a*** RAID ***storage device for storing data***, which includes disk arrays 1100A and 1100B. O'Connor fails to teach ***a controller for controlling data sent from said host through said data transfer network so as to assign said data to a virtual space and store said data assigned to the virtual space in said storage device***. Dang teaches a system for managing I/O request directed to a disk array (Dang, abstract). As shown in Dang FIG. 2 is a RAID storage device 26 with a plurality of disk array. As shown

in FIG. 1, when the processing unit 12 executes an instruction within the application program 19 corresponding to an I/O operation, control is transferred to the operating system 18. The operating system 18 creates a virtual disk I/O request corresponding to the I/O operation requested by the application program, and stores the virtual disk I/O request in RAM 14 as ***a virtual space***. A virtual disk I/O request can either be a virtual disk write request, for which data is to be written to the disk array 26, or a virtual disk read request requiring a disk array data read operation (Dang, Col. 6, lines 39-57). The request is broken into one or more sub-requests (Dang, Col. 7, lines 15-16), and stored in a pending queue (Dang, Col. 8, lines 14-15). If the request is a write request (Dang, FIG. 7A, step 204), the data are written into appropriate disk drives of disk array 26 as ***storage device*** (Dang, Col. 7, lines 44-53). As seen, the Dang technique indicates ***a controller for controlling data sent from said host through said data transfer network so as to assign said data to a virtual space and store said data assigned to the virtual space in said storage device***. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the O'Connor network file system by including a controller for assigning data to a virtual space before storing in storage device as taught by Dang in order to minimize the amount of time required for performing read write operation in RAID storage device of storage area network Universal File System.

Regarding to claim 6, O'Connor teaches a network file system and method wherein a storage area network Universal File System allows any host in a

heterogeneous based storage area network to read or write data as if in its native format (Abstract). As shown in FIG. 4 is a storage area network includes Universal File System (UVFS) storage devices 420, type A storage device 422, type B storage device 424, type A hosts 402 and 406, and type B hosts 404 and 408. Type A host 402 and type B host 404 both include universal file system mechanisms 410 and 412, respectively. The file system of the UVFS storage devices 420 is not compatible with the file systems of type A hosts 402 and 406, or type B hosts 404 and 406. Also, the file system of type A hosts 402 and 406 is not compatible with the file systems of type B hosts 404 and 408 (FIG. 4, Col. 5, lines 14-31). As seen, the O'Connor storage area network indicates ***a host for obtaining files from said storages***; ***a server for managing files present apart form said host***. To enable a host to utilize a universal file system, software may be installed as part of the operating system of a host, which allows it to mount the universal file system. Once mounted, data may be read from and written to the file system. When a client mounts a directory on a server, that directory and subdirectories become part of the client's directory hierarchy. Each platform may have its own enabling software package. With a universal file system, each platform need only create a package for accessing the universal file system and can be assured of being able to share data with other platforms, which are enabled in like manner (Col. 6, lines 23-38). As seen, once mounted to the universal file system by utilizing the software, a directory, subdirectory, or a file under a directory or subdirectory in a particular file system such as type A as ***a format specific to an operating system on said host***, is converted to a type that common to the UVFS file

system as ***generic format files having a format common to said storages***. In other words, the O'Connor software performs the function of ***a converter facility, included in said host, for converting files of a format specific to an operating system on said host into generic format files having a format common to said storages***. As shown in ***the data transfer networks*** of FIG. 4, UVFS mechanism 410 is configured such that type A host 402 sees data stored on the UVFS storage devices 420 as if it were stored in a format compatible with its own type A format (Col. 5, lines 47-53). A universal permissions scheme modeled after the Unix scheme is used in a universal file system to ensure data security and integrity. For example, when configuring a host for a SAN universal file system, a listing of the permissions mask for a file may be "Urwxr-x-x". In this case, the first character indicates this is a universal file system and should be treated as such. Each user may have one or more of the following permissions: read access, write access, or execute access. When a user attempts to access a file, the operating system first identifies which type of user is making the request, then checks the permissions for that user to determine if access is granted (Col. 6, line 48-Col. 7, line 18). Thus, by mounting the host file system on the UVFS and utilizing the software, a type A host 402 of FIG. 4 accesses a file had ***format specific to the operating system on said host*** in Unix scheme modeled UVFS 420 by conventionally specifying the file name such as "Urwxr-x-x" as ***a file name***, upon the file name reception, the process of retrieving of a file in UVFS, which has ***generic format***, is conducted by checking the permissions for that user to determine if access is granted, then displaying the file in a format compatible with its own type for

reading, modifying or executing. In short, this technique indicates ***server, connected to said data transfer network, receives a file name of a file in said format specific to the operating system on said host and manages transmission of said generic format files in response to an access permission request including said file name of a file in said format specific to the operating system on said host from one of said storages***. FIG. 12 is ***a*** RAID ***storage device for storing data***, which includes disk arrays 1100A and 1100B. O'Connor fails to teach ***a controller for allocating a data which is transferred through said data transfer network to a virtual space and storing said data allocated to the virtual space in said storage device***. Dang teaches a system for managing I/O request directed to a disk array (Dang, abstract). As shown in Dang FIG. 2 is a RAID storage device 26 with a plurality of disk array. As shown in FIG. 1, when the processing unit 12 executes an instruction within the application program 19 corresponding to an I/O operation, control is transferred to the operating system 18. The operating system 18 creates a virtual disk I/O request corresponding to the I/O operation requested by the application program, and stores the virtual disk I/O request in RAM 14 as ***a virtual space***. A virtual disk I/O request can either be a virtual disk write request, for which data is to be written to the disk array 26, or a virtual disk read request requiring a disk array data read operation (Dang, Col. 6, lines 39-57). The request is broken into one or more sub-requests (Dang, Col. 7, lines 15-16), and stored in a pending queue (Dang, Col. 8, lines 14-15). If the request is a write request (Dang, FIG. 7A, step 204), the data are written into appropriate disk drives of disk array 26 as ***storage device*** (Dang, Col. 7, lines 44-53). As seen, the Dang technique indicates ***a***

controller for controlling data sent from said host through said data transfer network so as to assign data to a virtual space and store said data assigned to the virtual space in said storage device. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the O'Connor network file system by including a controller for allocating data to a virtual space before storing in storage device as taught by Dang in order to minimize the amount of time required for performing read write operation in RAID storage device of storage area network Universal File System.

Regarding to claim 9, O'Connor teaches a network file system and method wherein a storage area network Universal File System allows any host in a heterogeneous based storage area network to read or write data as if in its native format (Abstract). As shown in FIG. 4 is a storage area network includes Universal File System (UVFS) storage devices 420, type A storage device 422, type B storage device 424, type A hosts 402 and 406, and type B hosts 404 and 408. Type A host 402 and type B host 404 both include universal file system mechanisms 410 and 412, respectively. The file system of the UVFS storage devices 420 is not compatible with the file systems of type A hosts 402 and 406, or type B hosts 404 and 406. Also, the file system of type A hosts 402 and 406 is not compatible with the file systems of type B hosts 404 and 408 (FIG. 4, Col. 5, lines 14-31). To enable a host to utilize a universal file system, software may be installed as part of the operating system of a host, which allows it to mount the universal file system. Once mounted, data may be read from and written to the

file system. When a client mounts a directory on a server, that directory and subdirectories become part of the client's directory hierarchy. Each platform may have its own enabling software package. With a universal file system, each platform need only create a package for accessing the universal file system and can be assured of being able to share data with other platforms, which are enabled in like manner (Col. 6, lines 23-38). UVFS mechanism 410 is configured such that type A host 402 sees data stored on the UVFS storage devices 420 as if it were stored in a format compatible with its own type A format (Col. 5, lines 47-53). As seen, once mounted to the universal file system by utilizing the software, a directory, subdirectory, or a file under a directory or subdirectory in a particular file system of a host such as type A, is converted to a type that common to the UVFS file system, and data in UVFS are displayed in the host as if it were stored in a format compatible with its own type A format for reading or writing. In other words this technique indicates ***a host having a file system for converting files in a file format specific to an operating system of said host into files in a format common to said storages, and converting files in said common file format on said data transfer network into files in said file format specific to said operating system of said host, and said host updating data in said file format specific to said operating system***. As shown in ***the data transfer networks*** of FIG. 4, UVFS mechanism 410 is configured such that type A host 402 sees data stored on the UVFS storage devices 420 as if it were stored in a format compatible with its own type A format (Col. 5, lines 47-53). A universal permissions scheme modeled after the Unix scheme is used in a universal file system to ensure data security and integrity.

For example, when configuring a host for a SAN universal file system, a listing of the permissions mask for a file may be "Urwxr-x-x". In this case, the first character indicates this is a universal file system and should be treated as such. Each user may have one or more of the following permissions: read access, write access, or execute access. When a user attempts to access a file, the operating system first identifies which type of user is making the request, then checks the permissions for that user to determine if access is granted (Col. 6, line 48-Col. 7, line 18). Thus, by mounting the host file system on the UVFS and utilizing the software, a type A host 402 of FIG. 4 as ***a host from one of said storages*** accesses a file, which had ***format specific to the operating system on said host***, in Unix scheme modeled UVFS 420 by conventionally specifying the file name such as "Urwxr-x-x" as ***a file name***, upon the file name reception, the process of retrieving of a file, or ***readout***, is conducted by checking the permissions for that user to determine if access is granted, then displaying the file in a format compatible with its own type for reading, modifying or executing. In short, this technique indicates ***a management facility, connected to said data transfer network, for receiving a file name of a file in said format specific to the operating system on said host and managing a readout of a file of said file format common to said storages in response to said file name of said file in said format specific to the operating system of said host from one of said storages***. As shown in FIG. 12 (Cols. 9-10) is a ***storage includes a file storage area for storing files in a format common to said storages***. O'Connor fails to teach ***a virtual space for retaining file that may be transmitted and received to and from said host or another storage and that is in said format common to said storages, and a***

storage controller for asynchronously allocating said file read out from said file storage area to said virtual space to transmit to said host said file in said virtual space. Dang teaches a system for managing I/O request directed to a disk array (Dang, abstract). As shown in Dang FIG. 2 is a RAID storage device 26 with a plurality of disk array. As shown in FIG. 1, when the processing unit 12 executes an instruction within the application program 19 corresponding to an I/O operation, control is transferred to the operating system 18. The operating system 18 creates a virtual disk I/O request corresponding to the I/O operation requested by the application program, and stores the virtual disk I/O request in RAM 14 (Dang, Col. 6, lines 39-51) As seen, the RAM 14 is ***a virtual space for retaining file that may be transmitted and received to and from said host or another storage***. Within the RAM 14 is data that will be transferred to or read from the disk array 26 (Col. 6, Lines 61-63). The virtual disk I/O request is broken into one or more sub-requests (Dang, Col. 7, lines 15-16), and stored in a pending queue (Dang, Col. 8, lines 14-15). As seen, by putting I/O request in a queue, files from RAID 26 as ***file storage area*** are ***asynchronously allocated*** into a queue and stored in RAM 14 as ***virtual space*** for transmitting to a host. In other words, the Dang technique of controlling the read request indicates ***a storage controller for asynchronously allocating said file read out from said file storage area to said virtual space to transmit to said host said file in said virtual space***. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the O'Connor network file system by including a controller for allocating data to a virtual space before transmitting as taught by Dang in order to minimize the

amount of time required for performing read write operation in RAID storage device of storage area network Universal File System.

Regarding to claims 2 and 5, O'Connor and Dang teaches all the claim subject matters as discussed in claims 1 and 4, O'Connor further discloses ***unit of data specific to said operating system has an actual data section and a first control section for defining the type of data specific to said operating system, said converter facility considers the entire unit as said actual data to add to said unity of data specific to said operating system a second control section created for managing the type of data and for being common to said storages*** (Col. 4, line 45-Col. 5, line 13; Col. 6, line 39-Col. 7, line 18).

Regarding to claim 3, O'Connor and Dang teaches all the claimed subject matters as discussed in claim 2, O'Connor further discloses ***data transfer network is a storage area network*** (FIG. 4).

Regarding to claim 7, O'Connor and Dang teaches all the claimed subject matters as discussed in claim 6, O'Connor further discloses ***a storage for storing said common format files, wherein said server issues to said storage a staging request with a file operation ID added with respect to a file requested for said access permission, and sends said file operation ID on condition that any error occurs; wherein said storage stages said file in accordance with said staging request and add said file operation ID to said file, and wherein said host obtains said file by issuing a***

file operation request to said storage with said file operation ID added (FIG. 12, Col. 4, line 45-Col. 5, line 13; Col. 6, line 39-Col. 7, line 18).

Regarding to claim 8, O'Connor and Dang teaches all the claimed subject matters as discussed in claim 7, O'Connor further discloses *file operation ID is for use in the acknowledgment of access right of said host* (Col. 4, line 45-Col. 5, line 13; Col. 6, line 39-Col. 7, line 18).

Regarding to claim 10, O'Connor and Dang teaches all the claimed subject matters as discussed in claim 9, O'Connor further discloses *data transfer network comprises a plurality of fibre switches having hosts and/or storage devices connected thereto and a storage area network for connecting these components* (FIG. 4, Col. 3, line 53-Col. 4, line 2).

Regarding to claim 11, O'Connor and Dang teaches all the claimed subject matters as discussed in claim 9, O'Connor further discloses *file in said file format specific to said operating system is comprised of actual data and a file control section for defining the file type thereof; and wherein said file system considers said actual data plus said file control section as an actual data entirely to create another file control section common to said storages, said file in said file format specific to said operating system being converted to a file in said file format common to said storage storages by adding said another control section to said file in said file format specific to said operating system* (Col. 4, line 45-Col. 5, line 13; Col. 6, line 39-Col. 7, line 18).

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to HUNG Q PHAM whose telephone number is 703-605-4242. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, JOHN E BREENE can be reached on 703-305-9790. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

6. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Examiner Hung Pham
July 13, 2004

SHAHID ALAM
PRIMARY EXAMINER